National Park Service
Database Specifications
for
Inventory and Monitoring Studies

DRAFT VERSION

Inventory and Monitoring Program
National Park Service

Summary

It is intended that inventory and monitoring studies will generate tabular data. To organize data across the park and region and to prepare it for summary at the national level, database design standards must be implemented. The primary goal of good database design is to ensure that data can be easily maintained over time. Standards improve a good design by allowing the development of consistent databases, which is extremely important when data are shared among multiple users, converted between multiple database servers and analyzed on multiple levels. This document describes the specifications required for acceptable delivery of tabular inventory and monitoring data.

This document provides general standards for tabular data. Data managers may require approval of the database design and other additional specifications. Please consult with the appropriate data manager for approval on desired digressions from these standards.

Deliverables

A descriptive document will be delivered to the national Inventory and Monitoring program or the appropriate data manager on CD-ROM (preferred) and/or by compressed (zipped) file, along with each complete and verified Microsoft Access database (.mdb) meeting the following conditions. The CD should be in ISO 9660 format to allow cross-platform use (.mdb files on the CD should follow the 8.3 file naming convention – see the Naming Standards section of this document for more information).

Required:

- The database contains the core tables from the database template, populated and with properly defined relationships to the study specific tables
- The database is normalized into multiple tables with properly defined and documented relationships
- The database table and field names meet the adopted naming standards
- Each table is defined with a primary key that uniquely identifies records in that table
- Each field is defined with a description
- Each field value contains raw data, with no formats applied
- A database model is included
- The data have been verified using one of the standard data verification methods

Strongly Recommended:

- The database is described in minimal or complete Biological Profile FGDC format either by inclusion in the on-line NR/GIS Metadata tracking database, a local Dataset Catalog database or other recommended metadata database (e.g. SMMS).

As appropriate:

- Each field, where appropriate, is defined by a domain of allowable values
- Each field, where appropriate, is defined with a display format and/or input mask
- Examples of field data collection form(s)

The items in the preceding list are expanded on in the following sections of this document. Please refer to the Recommended Database Procedures document for more complete details, and where appropriate, step-by-step instructions and examples.

Descriptive Document

A text or Microsoft Word document (i.e. the current NPS standard) describing the dataset will accompany each database submission to provide necessary information for understanding the data submittal.

Overview of descriptive document contents

- Contents of the CD or zip file (e.g. the Readme file – see below)
- Description of the project
- Location of the project study plan and work plan
- Project leader's name and contact information
- Principal investigator's name and contact information*****
- Data set contact's name and contact information*****
- Description of the database model (entity relationship diagram and data dictionary)
- Sensitive data issues, if appropriate
- Description of data verification methods and results
- Additional comments/documentation references, where appropriate

***** A copy of the metadata report (either from the on-line NR/GIS Metadata system, a local Dataset Catalog system or other recommended metadata database) may be included in lieu of these elements. Include the PI as the originator of the data set. Note the file name of this report in the contents section of this Descriptive Document.

The following example of a Descriptive Document for a park with alpha code "CODE" may be used as a template.

CODE_BirdSurvey_Readme.doc (or .txt)

A CD-R in ISO 9660 format contains the following file:

CODEBird.zip containing the following files:

- *CODE*_BirdSurvey_Readme.doc (this descriptive document)
- *CODE*_Bird_File_Names.doc (naming convention or codes used for file names - if applicable)
- *CODE*Bird.txt/.html/.sgml – FGDC metadata formats
- *CODE*Bird.mdb – MS Access database
- *CODE*_Bird_Data_Dict.doc (data dictionary for CODEBird database)
- *CODE*_Metadata.txt (minimal format metadata file)

CODEBird database contains survey sampling data from summer 2001 for ground nesting species in CODE. Surveys were conducted with modified transect sampling. Statistical analyses were produced with SysStat software.

None of the data contained in this data set is considered sensitive.

The project study plan and work plan are stored on the CODE data manager's workstation.
The database model is stored on the CODE data manager's workstation.

Project Lead:	Dr. John Smith, Ecologist CODE Headquarters 1234 Main Street Anywhere, ST 00000 999-555-1111 voice 999-555-1122 fax john_smith@nps.gov
Principal Investigator:	As above…
Data Contact:	Data Manager CODE GIS Office As above…

Two cooperators independently verified data after data entry by the principal investigator. Of the records in the CODEBird.mdb file, 0.03% had data entry errors which were subsequently corrected.

Additional comments as needed.

See the Data Modeling and Metadata sections of the Recommended Database Procedures document for details on data models and metadata creation options.

Descriptive Document Specification

Each database will be submitted with a descriptive document containing information about the project and data.

Database Template

The Natural Resources Database Template is a reusable set of tables and fields for storing inventory and monitoring data. By collecting common fields in all projects and standardizing all other fields, these data can be compared and analyzed together with similar data across the landscape. Common data can even be summarized and deposited in a national database.

The required core tables are common to all studies. They answer the basic questions: when and where were the data collected? Each piece of data has a location of collection along with characteristics that uniquely describe that point geographically. Similarly, each piece of data has a date and time of collection along with characteristics that describe the physical conditions at that point in time. The layouts of these tables are static and allow all collected data to be summarized at the program level. With few exceptions, these tables will be imported as is into the database and should not be modified. An additional core table contains one record with metadata about the project.

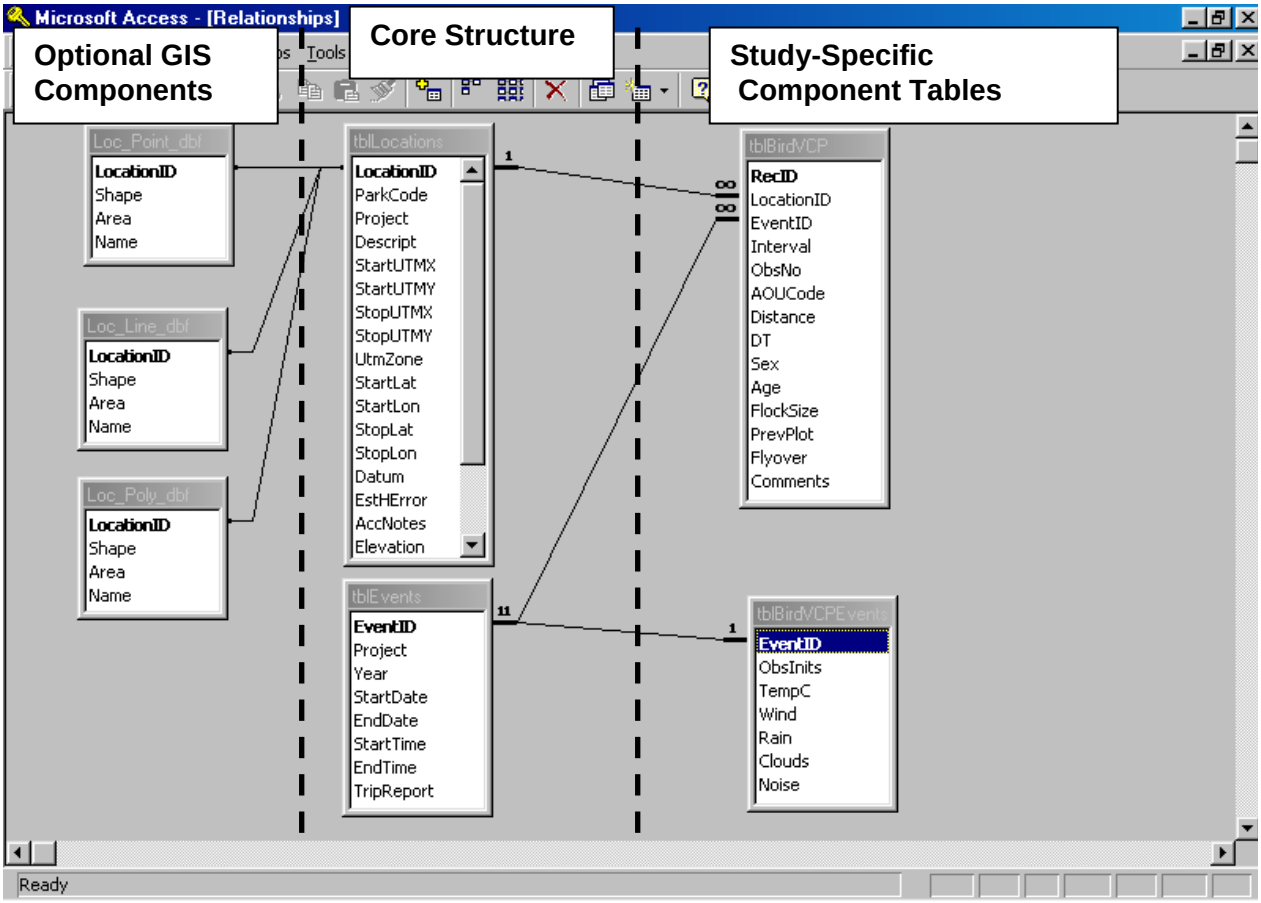
Over time, a supplemental set of optional tables specific to individual inventory and monitoring studies will be developed. These tables will act as libraries of sample fields for each type of study. Users are encouraged to use the appropriate library as a base for their data model. These tables are dynamic; they will become more robust as users append additional useful fields.

The database must contain a unique identifier or key for each data collection site. This key, along with geographical references, will be stored in the core site location table (e.g. LocationID in tblLocations). The database must also contain a unique identifier for each observation or sampling event. This key will be stored in the core event table (e.g. EventID in tblEvents). Using these keys, all other tables will be related, directly or indirectly, to one or both of these core tables. Using unique keys to relate tables is a primary component of data normalization, which is discussed in the next section of this document.

Overview of Database Template Implementation

- Import the most recent version of the core Database Template tables. The Database Template database files are available from the I&M web page under the Applications heading.
- Working with the appropriate data manager, design the remaining study-specific tables around, or related to, the Database Template as defined by the study plan (see the figure below for an example design). Relate study-specific data tables to the core tables using the tblLocations.LocationID and tblEvents.EventID keys.
- Create field collection forms and complete the field study.
- Populate the database with the field data, including the core Database Template tables.

Core Database Template Structure with Study-Specific Tables:



(adapted from Database Template Phase 2 Documentation – October 9, 2002)

Please refer to topic Set field properties to customize how data is stored, handled, or displayed in the Microsoft Access Help file for more details on field creation. The Data Modeling and Data Normalization sections of the Recommended Database Procedures document contain information on documenting a database model, creating data dictionaries and normalizing a database.

Database Template Specification

Each database of field data will contain the core database template tables. All other tables will be related to one of the core tables.

Data Normalization

Data normalization is the optimization of database tables. Normalization separates the fields from a large, generalized table into multiple, smaller related tables by removing all unnecessary or duplicate fields and ensures that each table represents a single subject. This includes the process of converting data from a flat file (i.e. spreadsheet) into a relational database.

The overall goal of normalization is to remove ambiguity. This process can be broken down into six steps. The first three steps normalize the database into Third Normal Form. At this stage, data can be efficiently accessed and manipulated without the model becoming too decomposed and difficult to understand.

Overview of Third Normal Form

- Each table contains data about a single subject.
- Each table is identified with a primary key.
- No table contains repeating fields.
- No table contains redundant data, or groups of repeated values for multiple records.
- Each table contains only fields that are dependent on the primary key, or directly related to the subject of the table.

Additional information on data normalization can be found in the Data Normalization section of the Recommended Database Procedures document. Please refer to topic Set field properties to customize how data is stored, handled, or displayed in the Microsoft Access Help file for more details on field creation.

Data Normalization Specification

Each database will be optimized to Third Normal Form (3NF).

Naming Standards

Standards are an important part of database design, as they allow for the development of consistent databases. This is extremely important when data are shared among multiple users and converted between database servers. Table and field names should be designed to clearly define the data being stored. These names should be meaningful to the entire organization.

Table names will have the format PRE_ROOT where PRE is a three-character table prefix and ROOT is the root name. Many field names will have the format ROOT_SUF_UOM where ROOT is the root name, SUF is a defined field category, and UOM is an abbreviation for the unit of measure. In general, the maximum table and field name length should be around 20 characters. However, it is strongly recommended that spatial data or attribute data which could be imported into or linked to GIS or other PC database software (ArcView 3.x, dBASE, etc.) use 8 or 10 character maximum table and field names, respectively. In this software, table and field names longer than 8 or 10 characters will be truncated upon import, potentially sacrificing information by resulting in duplicative or unclear table and field names.

File names for database files should conform to the 8.3 standard to allow maximum cross-compatibility. Many software packages (including ArcView3.x and many PC databases) default to an 8.3 convention when importing or linking to files from MS Access. Also, the NPS currently has many network and other resources that utilize only 8.3 file naming.

Overview of Table Names

- Prefix. The prefix identifies the data object type (e.g. tbl for a data table).
- Root Name. The root name is a noun or short phrase that clearly defines the table (e.g. Voucher).

Overview of Field Names

- Prefix. The optional prefix identifies a field type as a boolean (yes/no) (e.g. Is_Present, other prefixes include ‘are’ and ‘has’).
- Root Name. The root name is a noun or short phrase that clearly defines the field (e.g. Event).
- Suffix. The suffix identifies the field category (e.g. Event_ID, Park_Count, Start_Date).
- Unit of Measure. The optional unit of measure abbreviation defines the required field unit (e.g. transect_length_km).

Overview of File Names

- For database files, utilize 8.3 file name convention whenever possible. 8.3 means 8 characters preceding the ‘.’ with a 3 character extension after the ‘.’.
- Do not use spaces or dashes in any file name.

Please refer to the Recommended Naming Standards section of the Recommended Database Procedures document for the specific details of the naming conventions, along with examples for each table type and field category.

Naming Standards Specification

Each table and field name will match the corresponding standard format and adhere to the standards defined for root names. Database file names will adhere to the 8.3 file naming convention.

Primary Key

The primary key of a relational table uniquely identifies each record within the table. Each table needs a primary key so that a single row can always be accessed or modified without altering any other records in the table. The values that compose a primary key must be unique; no two values can be the same. The primary key field(s) will always be required; no value can be null. The primary key can be a single field that is populated by the user or auto-generated by the system. Two or more fields will sometimes comprise the primary key, in cases where only the concatenation of multiple values forms a unique combination.

Any field or group of fields that is eligible as a primary key (i.e. will have a unique value) is called a candidate key. A table can have any number of candidate keys. One candidate key is chosen as the primary key and the remaining become alternate keys. Candidate keys may be noted in the field descriptions.

Please refer to topics Working with Primary Keys and Indexes and Set or change the primary key in the Microsoft Access Help file for more details on creating a primary key. Also, the Data Normalization section of the Recommended Database Procedures document details defining primary keys.

Primary Key Specification

Each table in the database will be identified with a primary key.

Field Description

A field description includes a definition statement that clearly states the purpose of the field. The description can be used to further clarify any information about the field that may not be apparent by the field name alone. Especially when data are shared among multiple users, it is extremely important to write clear and concise field descriptions. These definitions are documented in the database during table creation. Data dictionaries typically contain these field definitions, along with field data type information.

Please refer to topic Add a field to a table in design view in the Microsoft Access Help file for more details on creating a field definition. The Data Modeling section of the Recommended Database Procedures document discusses field descriptions in the context of the data dictionary.

Field Description Specification

Each field in the database will be defined by a clear and concise description.

Data Storage

A database is most efficient when populated with raw, unformatted values. Since formatted data can only be saved in text fields, formatted numeric and date values must be converted to text from their native data type. Users of the database lose the benefits of the numeric and date data types, specifically the calculations and functions that can be performed on those fields. Data entry time increases when formatted text is keyed in. Additionally, data formats are more difficult to control and cannot easily be modified without updating each record individually. Sorting performed on formatted values is not reliable. Special characters are taken into account during the ordering. But, more importantly, a sort performed on numbers or dates stored in a text field will return unexpected results, since the values are ordered by the ASCII code of the individual characters rather than the value as a whole.

There are two areas where formatting may be acceptable. Tables often contain description or comment fields. These fields are defined with text (maximum of 255 characters) or memo data types and contain written verbiage, often in paragraphs. Any formatting that is embedded within this style of text is allowed. Also, efficiency or security requirements may necessitate storing calculated fields (i.e. trading storage space for speed or storing single values in a secure table, but calculated values in a non-secure table). In the second case, consult with the data manager for approval of the calculated fields. Be sure such fields are documented clearly in the field description and data dictionary.

Data Storage Specification
Formatted text will not be stored in the database.

Required Field

Fields in a table for which values must be entered are mandatory or required fields. Since the primary key cannot contain a null value, the field(s) comprising the primary key will always be required. Other required fields should be identified, as well in the data dictionary.

Please refer to topic Properties that control how blank fields are handled in the Microsoft Access Help file for more details on designating a required field. In Oracle, required fields are defined as NOT NULL fields at the time of table creation. See the Data Modeling section of the Recommended Database Procedures document for more information.

Required Field Specification
Each table in the database will be identified with required fields.

Field Domain

The domain of a field is the set of all permitted values for that field. Defining a domain is important to restrict the entry of invalid data. The domain is tied into the field definition by the use of a validation rule (not recommended for shared data fields) or a picklist (derived from a reference table or lookup table).

Overview of Domains

- Unrestricted data type. An unrestricted data type domain allows the broadest range of values. Any value that is acceptable for the given data type is permitted in the field. The data type is identified during the table creation. For this domain type, no further steps are required on behalf of the designer, as the system automatically rejects any value that does not fit within the boundaries of the data type.
- Character set. A character set domain defines the allowable characters that are acceptable within a text field. These values should be defined in a reference or lookup table. Before accepting a value in this field, the system uses the domain to ensure that either the value entered passes the defined rule or the value does not include any of the restricted characters. This domain type is often implemented by restricting certain numbers, letters, or special characters from the field.
- Value range. A range domain defines a range of values that is permitted into the field. A range may be open-ended or close-ended and inclusive or non-inclusive. These values are defined by a validation rule for the field during the table creation. Before accepting a value keyed by the data entry person, the system makes sure that the value entered passes the defined rule, or the value falls within the allowable range (e.g. a pH between 0 and 14).
- Value list. A list domain defines a list of actual values that is permitted in the field. These values should always reside in a reference or lookup table that is related to the main data table. The list is generally displayed as a combo box (i.e. drop-down pick list box) defined as a lookup for the field during table creation.

Since all fields must be assigned a data type, the data type domain will place initial limitations on the values in every field. A formal validation rule or lookup should be defined to further restrict the allowable values for a field.

Note that validation rules are dependent on the database software used. Microsoft Access allows validation by domain definition at the time of table creation. Oracle implements these restrictions both during table creation and as external triggers and functions. One limitation of Access is in the definition of value lists based on another data (lookup) table. If the underlying lookup data table structure changes, the impacts on the dependent combo box(es) must be diagnosed and corrected. Otherwise, these types of errors become maintenance issues that are difficult to track and fix. Validation rules should be documented in the field definitions of a table and in the associated data dictionary document.

Please refer to topics Restrict or validate data or Work with Lookup fields in the Microsoft Access Help file for more details on creating a field domain. See the Field Formatting section of this document for more discussion of field content and format restrictions.

Field Domain Specification

Each field in the database will be identified by an appropriate domain. Where applicable, validation rules will be employed and documented.

Field Formatting

Since formatting is not permitted on values stored within the database, there are alternatives that allow data to be displayed in a formatted style.

Overview of Formatting Options

- Display format. For viewing and printing purposes, a display format may be defined to customize the way Boolean fields (yes/no), numbers, dates, times, and text are displayed in a text box. The special characters used in formatting are not stored as part of the value within the database, but simply are applied to the value every time it is displayed.
- Input mask. An input mask is similar to a format, but actually presents an empty format "shell" to the user during data entry. This mask forces the user to populate certain elements within the format. These elements may be restricted to allow only numbers or letters. An input mask helps control the data that is entered in a text box. An input mask should never store the formatting along with the field value; this option is offered during the input mask wizard.

In Microsoft Access, formatting definitions are produced during table creation. In Oracle, table creation statements, triggers and standard and custom functions are used to restrict input and display specific formats.

Please refer to topics Should I use a data display format or an input mask?, Define the data display format for a field in table Design view or Define an input mask for a field in a table in the Microsoft Access Help file for more details on formatting a field.

Field Formatting Specification

Each field in the database may be assigned formatting options for input and/or display.

Data Verification

Manual effort is generally required to get data into electronic format. Any errors made during typing will accumulate in the permanent database unless the data is verified and errors are detected. By implementing a data verification practice, these errors can be reduced, if not eliminated.

Overview of Data Verification Methods

- Visual review at data entry. The data entry person verifies each record after it is input. The values recorded in the database are compared with the original values from the hard copy and any errors are corrected immediately. This method is the least complicated since no additional personnel or software is required. The accuracy of this method depends wholly on the person keying data and is generally the least reliable of the data verification methods.
- Visual review after data entry. All records are printed upon the completion of data entry. The values on the printout are compared with the original values from the hard copy. Errors are marked and corrected in a timely manner. The reliability of this method increases if someone other than the person keying data performs the review. Again, no special software is required.
- Duplicate data entry. The data entry person completes all data input, as normal. Random records are selected (every **n**th record) and entered into an empty replica of the permanent database, preferably by someone other than the person keying the permanent data. A query is run to automatically compare the duplicate records from the two datasets and report on any mismatches of data. These disparities are manually reviewed and corrected if necessary. This method involves the overhead of re-typing the selected records, as well as the creation of a comparison query (which requires additional effort, but is not time-consuming). This method becomes increasingly successful as the value of **n** decreases.

Each method has a direct correlation between effectiveness and effort. The methods that eliminate the most errors can be very time consuming, while the simplest and cheapest methods will not be as efficient at detecting errors.

For more information, refer to the Data Verification Procedures and Data Validation sections of the Draft Natural Resource Information Management Framework Handbook.

Data Verification Specification

The data in each database will be reviewed and corrected using an approved data verification method, such that data accuracy is 95% or greater.

Data Verification Reporting

A description of the verification method and results will be included in the Descriptive Document accompanying the database.